JOHN PRICE-WILKIN

Systems Librarian for Information Services
Alderman Library
University of Virginia
Charlottesville, Virginia

The Feasibility of Wide-Area Textual Analysis Systems in Libraries: A Practical Analysis

ABSTRACT

This paper discusses the textual and software resources necessary for the establishment of a generalized wide-area textual analysis system. A distinction is made between textual analytical systems and text retrieval systems. The necessity of using standards and open systems in implementing such systems is emphasized. The paper includes a review of critical characteristics of generalized analytical software. It is argued that the resources necessary for the establishment of a service are currently available. The paper concludes with a discussion of deficiencies in current resources and standards. The author also includes an appendix discussing the need to incorporate a recognition of structure in textual retrieval systems.

INTRODUCTION

I propose to offer an assessment of where we stand in being able to offer wide-area access to textual analysis resources based primarily on my experience in providing support for wide-area textual analysis systems. I will begin by defining what I believe is necessary for the establishment of a wide-area electronic text service, including what we mean by a service that supports textual analysis, and will discuss the relevance of open systems and standards. I will give some attention to the availability of textual resources—commercially and informally

distributed—and will provide a lengthier discussion of the capabilities of textual analysis software needed to take advantage of standardized encoding. And finally, I will briefly discuss the lack of standard mechanisms for access and protocols for search and retrieval.

MODELS AND CONFUSION

Computer-aided analysis of text has a relatively long history, but only in the last few years have we established access mechanisms at the institutional level. There is now a relatively young and promising situation for wide-area support for textual analysis. There are a few widely divergent models for providing resources to communities of scholars, and there is confusion in the marketplace about what resources are appropriate for the analysis of text. Because this discussion is only a consideration of wide-area networking of the resources of textual analysis, I will not consider those cases where the support consists of a text center where the actual work with texts and software is performed. With that consideration in mind, I believe there are three examples. They are ARTFL (American Research on the Treasury of the French Language) at the University of Chicago, the systems developed by Malcolm Brown at Stanford and Dartmouth, and the service offered by the University of Michigan and later expanded at the University of Virginia. Rather than explore each system exhaustively, I will highlight aspects of the three to define the context of this discussion of wide-area textual analysis services. The three models represent different approaches to how materials are accessed, the collections offered, and the encoding of those collections.

ARTFL is probably the first example of a system that offered immediate access to its collection of text processed for access with an analytical system. In 1988, ARTFL moved from offline access to texts and developed their own UNIX search engine and encoding scheme to provide access to their body of some 3,000 titles. While ARTFL's software can also support client/server transactions and is not tied inextricably to the interface with which most of us are familiar, its strategy is to centralize its corpus and provide primary access to the materials through their PhiloLogic interface. Libraries interested in offering access to ARTFL's collections are not burdened with issues such as transforming ARTFL's encoding scheme to meet local needs and deciding on methods of indexing or subsequent encoding. Similarly, they are also not able to define collections in ways that suit local needs by making texts available for use with other software packages or adding markup to facilitate different analytical approaches.

Malcolm Brown led the development of a server protocol and graphical client first at Stanford and then at Dartmouth to gain access to those universities' collections of texts. In both cases, the systems use Open Text's PAT search engine[1] and, for its texts, markup suggestive of Standard Generalized Markup Language (SGML). The client was developed in 1990 at Stanford as Searcher and was elaborated as part of an entire system of information retrieval (Dartmouth College Information System—DCIS) at Dartmouth (Brentrup 1993). The systems at these institutions are oriented toward a protocol that has grown to be an extension of Z39.50. Collections are limited, and development efforts are devoted to effective clients that serve general needs, offer a polished graphical interface, and provide a substantial degree of reliability. Markup is limited to that necessary for the presentation and basic functionality of generalized queries. Specialized access, it could be argued, is not formally supported.

The implementations at Michigan and Virginia differ from the ARTFL and Stanford models in their focus on building SGML-compliant collections and the minimal attention they devote to the development of an interface or client for the resources. They also use PAT. At Michigan, most work was done with command-line access to the PAT software itself, and only later was a vt100 client introduced.[2] Both institutions rely primarily on the Open Text clients but support all clients that can query the PAT search engine, including World Wide Web (WWW) forms-compatible clients such as Mosaic. Collections are actively expanded through traditional collection development activities. At the University of Virginia, the process of applying markup is done collaboratively with Electronic Text Center staff, systems staff, and catalogers, and in almost every effort, the Text Encoding Initiative (TEI) guidelines are used for that process. Work with researchers at both institutions guides the application of markup and indexing for specialized purposes. All of this effort in the construction of texts is reflected in the materials offered through the wide-area services.

A brief digression is useful at this point to consider a resource that contrasts to the initiatives already noted. Chadwyck-Healey markets its texts in two ways: it sells the texts themselves, with SGML, on tape, and it sells the texts on CD-ROM formatted with a modified version of EBT's DynaText software. Both Chadwyck-Healey's textual initiative and EBT's DynaText are laudable—Chadwyck-Healey for undertaking a project with immense potential impact on scholarship in the humanities, and EBT for creating an excellent SGML publishing tool. The combination of these two resources, however, reflects a significant degree of confusion about the needs of textual analysis. EBT's software is

designed for publishing electronic texts and does not have significant analytical capabilities. Its sophistication in browsing and formatting information is excellent, and it also provides search capabilities that exploit the SGML tagging. While it is well suited to supporting a document retrieval system, it does not give scholars the ability to easily examine large numbers of occurrences or to search quickly across large collections. The misapplication of DynaText to this project is ironic in that the *English Poetry Database* is perhaps the most substantial resource for literary computing made available to date and needs to be supported by an appropriate analytical engine.

Variation in the implementations of analytical systems is not a bad thing: it means there are choices. The status of software at the heart of the three models discussed a moment ago is perhaps a further indication of the infancy of providing these sorts of services. The software at ARTFL is an in-house effort and is not yet available for implementation outside that institution. The other two models use PAT as their search software: this software is marketed by Open Text as a general-purpose search "engine" and is sold without any apparent awareness of or guidance in how access will be provided to the textual resources. Despite failings in all three of the models, each is an excellent foundation and stands in contrast to the CD-ROM solution provided by Chadwyck-Healey, which simply does not offer tools for analysis. Each is a model of wide-area textual analysis because it provides access to large bodies of texts, facilitates a broad range of analytical functions, and is remotely accessible to all platforms used at those institutions. Two of the three are models that can be easily implemented at other institutions.[3]

Despite this minor chaos, there is an atmosphere of enthusiasm. University libraries around the United States are establishing electronic text centers; Chadwyck-Healey continues to announce the publication of new series in SGML; and rumors of academic text creation projects emerge periodically. The optimistic atmosphere results in part from momentum: the establishment of an increasing number of text centers encourages more people to see this as a viable movement, thus encouraging more university libraries to establish electronic text centers. Other factors are playing an important role: the creation and leadership of the Center for Electronic Texts in the Humanities (CETH) and the development of the TEI guidelines are two profoundly significant factors. Also playing a role is the availability of appropriate software to facilitate services and the growing wealth of textual resources. These are some of the more obvious factors that I see lending weight to a growing mainstream acceptance of computer-aided analysis of text and the development of initiatives for wide-area networking of the resources. Nevertheless, the notion that adequate texts and software are available

is not generally accepted. I would like to proceed to examine the nature of these assumptions, beginning by explaining what I see as necessary minimal resources for textual analysis.

TEXTUAL ANALYSIS, NOT ELECTRONIC PUBLISHING

I risk stating the obvious by saying that textual analysis stands in contrast to electronic publishing with tools such as DynaText and full-text electronic document delivery as seen in Project Mercury or TULIP.[4] While electronic publishing projects deliver a fully integrated, ready-to-read *electronic document,* and efforts such as TULIP are designed primarily to expeditiously produce facsimiles of articles or books, textual analysis focuses on computer-aided processes that aid in determining characteristics of text. This is not a case of bad and good approaches: the software used in textual analysis, document retrieval, and electronic publishing will almost necessarily be very different. For example, a textual analysis system must support phrase searching as easily as it does a word search. The notion of stop words is untenable. Absolute precision in retrieval is essential, and probabilistic methods as found in software like Wide Area Information Servers (WAIS), Topic, and Smart are unlikely to be useful. Truncation cannot significantly increase the amount of time needed to retrieve results. And while document retrieval systems return large chunks of text by design and can provide key words in context (KWIC) displays only with difficulty, fine-level results such as the KWIC are a fundamental part of the textual analysis system. These same characteristics may be found in the other systems, but for textual analysis, they are critical.

Consider the following example of textual analysis using the example of the vowel shift in English, e.g., the change from lond to land and lomb to lamb. In all of Old English, there are 262 works that contain the stem "lond" and 1,239 works that contain the stem "land."[5] Only 126 works contain both stems, and in those 126 works, there are 834 occurrences of the stem "lond" (Figure 1a). This is an interesting result for an Anglo-Saxonist. The person who, looking at the complete Old English Corpus, wanted to see all 834 instances of the string "lond" in texts that also have the string "land" almost certainly does not want much broader context than the relevant lines, and definitely does not want all 126 works delivered to his printer so that he can read them later at his convenience. Eight hundred thirty-four is only a moderately overwhelming number of occurrences, but let us expand the problem to include all "on/om" and "an/am" strings. Looking at the more than 300,000 relevant occurrences will be quite a task even in a system that provides KWIC displays or displays of

relevant lines, but a system that can further refine the results based on other textual characteristics (e.g., verse works as opposed to glosses) can make this manageable. Figures 1b to 1d represent the results of narrowing the search based on textual characteristics such as genre, language, and period. The raw search results, all taken from the entire Old English Corpus, were retrieved in a total time of less than two seconds. The method and results are not uncommon for a textual analysis system.

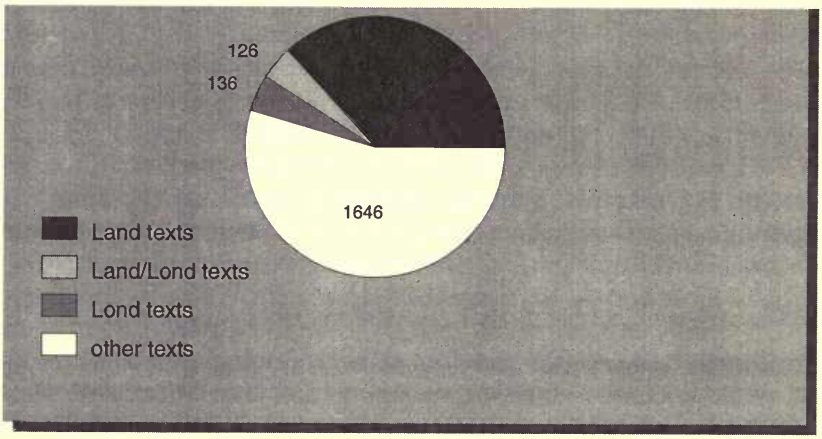

Figure 1a. Distribution of lond and land in Old English Corpus (area of intersection [126 texts] includes 834 occurrences of "lond")

STANDARDS AND OPEN SYSTEMS

Standards and open systems approaches must be a defining part of these efforts to provide the resources of textual analysis to communities of scholars. It is not enough to say that access is improved if a major investment is made in textual resources that will be unusable in two years. The texts must be reusable. It must be possible to use the texts in a variety of types of analysis, with a variety of analytical packages. Additionally, the texts must be accessible to a variety of computing environments. Because of the cost of creating the texts, investing in the texts must be an investment in the future. Selecting texts based on a system's capabilities when that system excludes the possibility of simultaneously using the texts with other tools is to restrict the field of inquiry. To that end, a standards-based encoding scheme and a generally agreed upon tag set must be at the foundation of text creation.

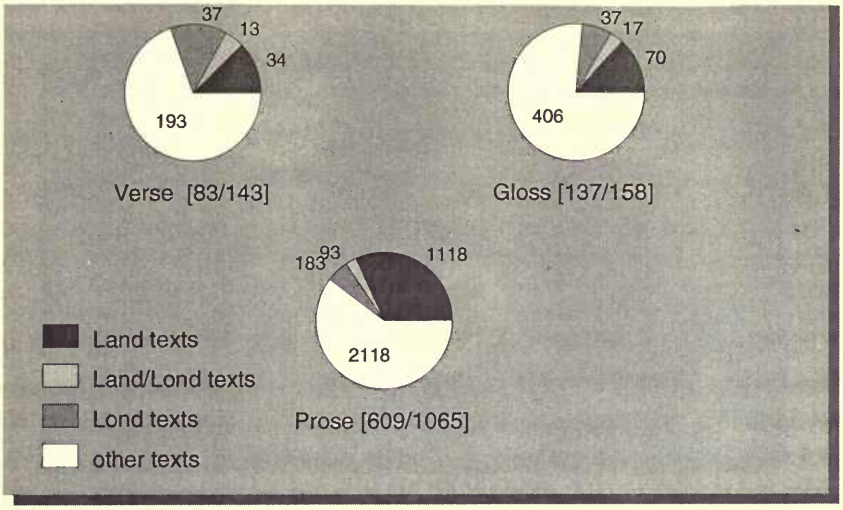

Figure 1b. Distribution of lond and land in Old English Corpus subset (number of "lond/land" occurrences in area of intersection listed in brackets)

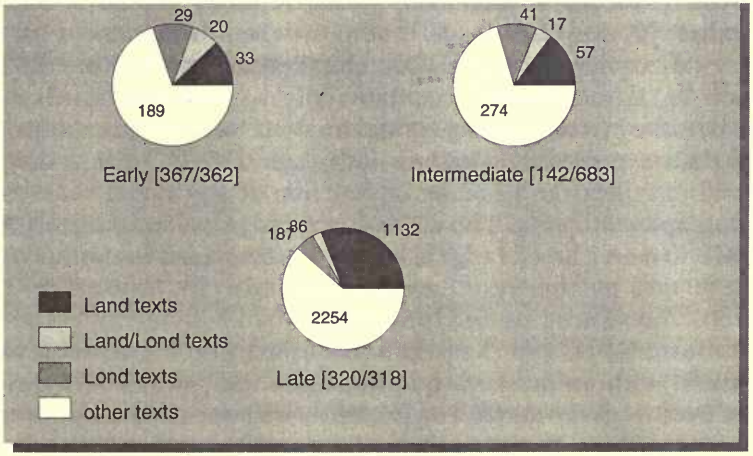

Figure 1c. Distribution of lond and land in Old English Corpus subset (number of "lond/land" occurrences in area of intersection listed in brackets)

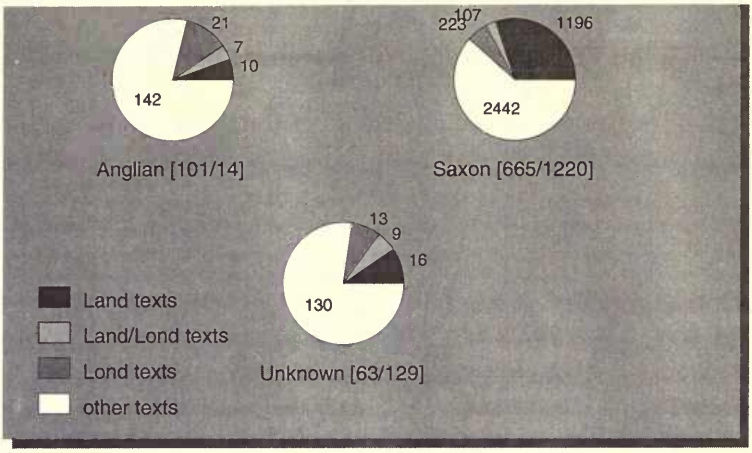

Figure 1d. Distribution of lond and land in Old English Corpus subset (number of "lond/land" occurrence in area of intersection in brackets)

The application of SGML through the Text Encoding Initiative will continue to play a central role in ensuring that resources are produced in a way that makes them flexible and of continuing value. This paper is not the place for an argument of the value of ISO 8879, Standard Generalized Markup Language, especially when the argument has been made so effectively elsewhere (Coombs, Renear, and DeRose 1993). In addition to its value as an internationally approved standard, SGML is ideally suited to supporting textual analysis because it is a descriptive rather than a procedural markup language. That is, it is a language designed to reflect the structure or function of text rather than simply its typography or layout. The difficulty of designing an implementation of SGML to meet a broad range of text-processing needs in the humanities has been met by the Text Encoding Initiative in its *Guidelines for Electronic Text Encoding and Interchange*.[6]

It must also be said that text without markup is irrelevant to this discussion: without markup, questions like the "lond/land" question cannot be effectively asked. For example, without markup, there is no effective, standards-based way of representing the "body" of the text and distinguishing it from descriptive information about the text. There are some alternatives to markup, including, for example, modeling the text's structure through a database or (as HTML and Gopher do) through the use of the filesystem. Reliance on a database (instead of encoding) for representing structure is inadequate because of its proprietary format.

Use of the filesystem—directories and files—to represent structure is quickly overwhelmed by complexity and the amount of information found in thousands of documents.

By using open systems to support access to materials, the textual analysis system can support the widest variety of platforms. An integral part of this is the establishment of access standards for textual analysis. Z39.50 is a capable mechanism for access to bibliographic information, but it will almost certainly fall short when dealing with complex documents and structural relationships (see the Appendix). Short of such a standard, however, an effective strategy is the use of a published or documented protocol. The PAT query language is at the foundation of the Telnet-derived protocol in use at the University of Virginia. Using this strategy, the university has been able to support wide-area access to a commercial X-Windows client (PatMotif), a locally developed vt100 client (URL: file://etext.virginia.edu/pub/clients), a commercial MS-Windows client (PowerSearch), WWW forms-compatible clients such as Mosaic and OmniWeb, a student-developed X-Windows OED client, and the PAT command-line.

WHAT TEXTUAL RESOURCES?

Our needs for textual resources are at least as great as our historical collections, but we have begun to enjoy some benefits from standards efforts and the increasing interest in electronic publishing. The body of material available from the Oxford Text Archive, parsed and in TEI-conformant markup, grows as a result of current efforts in creating new texts and the efforts of Jeffery Triggs,[7] the University of Virginia, and others in the conversion of previously deposited materials. In addition to the Old English Corpus (a complete representation of Old English assembled for the *Dictionary of Old English*), more than 100 works of some quality are available from the Oxford Text Archive, most via anonymous File Transfer Protocol (FTP).[8]

Commercial offerings have begun to have some impact on the collections we build. Chadwyck-Healey's often controversial offerings comprise the largest portion. Their *English Poetry Database* is projected to be completed in late 1994, and already we have more than 1,500 volumes of English verse as a result of it. Forthcoming projects include an English verse drama series, an African-American verse (to 1900) series, and a recently announced *American Poetry Database*. Their *Patrologia Latina Database,* including the 200 volumes of Migne, should also be finished this year. These efforts are notable for the scrupulous (if generic) application of SGML. Also producing SGML texts is Oxford University Press, with a varied and attractive publication list. The University of

Michigan Press will begin publishing works in SGML from the Society for Early English and Norse Electronic Texts (SEENET), late this year. Though not specifically in SGML, high-quality philosophical texts are being published by INTELEX; the University of Virginia is applying SGML to many of these and returning the marked-up versions to the publisher for subsequent resale.

It is remarkable that by the end of the year it will be possible to offer nearly all of England's verse online,[9] to offer all of extant Old English, and to offer significant bodies of Middle English materials. If literary research in the electronic environment has suffered from being limited by the body of material available, this body of "extraordinary language" (as contrasted to the "ordinary language" of everyday speech and writing) should begin to change that. But there are many questions about the quality of the materials.

There are several common criticisms of electronic texts: many electronic texts are based on poor editions, some are poorly transcribed, and (of the ones with SGML) the level of markup will not support the most sophisticated investigations. Many of the criticisms are appropriate, but the problems with many of the currently available electronic texts are opportunities rather than failings.

The choice of editions is frequently made more problematic by issues of copyright, but poor editions have always been part of our collections and have played a role in defining good editions. Scholars working on new editions at the University of Virginia have expressed the hope that electronic texts drawn from poor editions could serve as the foundation for current editorial efforts. Some of the available electronic texts are from editions of the highest quality. Several scholars have contributed the files used in creating scholarly editions (for example, Frances McSparran, who contributed the files for her Early English Text Society *Octovian*). One significant publisher, Library of America, has contributed typesetting tapes of several titles to the Oxford Text Archive. In these instances, we have only the most authoritative sources for exceptional editions.[10]

Limited markup is an opportunity for many interested in more sophisticated forms of analysis: detailed markup is usually a function of a particular type of analysis and so is unlikely to suit all users. One of the advantages of working with standards-based and generally agreed upon markup is that it is possible to build on the work of others. It will be difficult to identify or define a minimum level of markup for electronic texts, but a foundation of structural markup for commonly recognized features (e.g., poems, stanzas, and lines in a volume of verse) will serve most needs.

It is also true that there are poorly transcribed texts that should not be archived. We have received files that the depositor explains are failed experiments with scanning that even the scholar who produced them was not willing to use. Poorly transcribed texts pose an unambiguous threat to scholarship and should be clearly identified and set aside so that they may be used only by those fully aware of the problems inherent in the texts.

A bigger problem than the quality of the electronic texts is the current conditions of use in commercial resources. For example, even with the most liberal license, it is not possible for a scholar to begin work creating a new edition of a work using an electronic edition from a commercial publisher. More typically, it is even difficult to use a given text in two environments (e.g., with a statistical program and a campus-wide textual analysis system) simultaneously. The cost of acquiring these collections along with licensing restrictions is creating a situation where only those affiliated with larger, well-endowed universities have access to these resources. In general, these problems are creating a disjuncture with traditional roles of research libraries, as the libraries are no longer able to serve the role of augmenting the collections of smaller institutions, and the resources acquired cannot be used in the continuum of scholarship where older editions form the basis for newer editions. The economics of publishing are likely to ensure that these situations do not change. We should begin to look at alternative strategies for creating electronic texts for libraries, including creating the texts ourselves.

CONDITIONS NEEDED FOR A WIDE-AREA ANALYTICAL ENVIRONMENT

A system for textual analysis will necessarily be more complicated than a simple document retrieval system because of the operations that are performed. Operations include those mentioned earlier, such as effective phrase searching, easily browsed large result sets, and precision in searching. In addition to those capabilities, a textual analysis system should include other features:

- *Efficient cross-textual analysis:* The system should not constrain the user to searching a single text or a small group of texts with each search. It should be possible to search large bodies of material and quickly get a response (i.e., within seconds).
- *Expeditious results:* Similarly, most searches should yield results in seconds, not minutes. A search of a truncated stem "lond" in the *English Poetry Database* at Virginia takes approximately one second. The same search takes several minutes on the Chadwyck-Healey

CD-ROM, a problem attributable to both the DynaText software and the organization of data on CD-ROM.

- *Recognition of structure:* Software must be able to examine structural relationships and, specifically, to locate occurrences of features defined by their structural placement. (What do we mean by structure? The organization of a text explicitly and implicitly makes clear elements of structure such as chapters, paragraphs, poems, stanzas, and verses. These more obvious elements are often signaled to the reader in unambiguous ways by headings, but just as frequently, as with stanzas, they are indicated by conventions generally recognized. Structures may also be composite or abstract, as in "16c quotations in the OED," where the era of the quotation is signaled by a feature of a substructure, i.e., date.) For example, to examine the vowel shift in the Old English Corpus accurately, it is necessary to eliminate both the Latin text within the Corpus and the descriptive information accompanying the texts. To identify rhyme or other line-end features, it must be possible to distinguish between text that appears before a carriage return (e.g., in prose formatted to display on the screen) and a true end-of-verse.

In addition to these more common forms of analysis, other activities such as morphological analysis, statistical analysis of occurrence patterns, and general grammatical analysis—i.e., the recognition of constructions of various types—will need to be supported. We should expect to see at least as many forms of computer-aided textual analysis as we would find in print textual analysis.

A single package will not accommodate all sorts of analysis but can satisfy a majority of the fundamental needs of textual analysis. Understanding this leads us to other important conclusions. For example, the importance of reusable text, of text whose signals are rendered in a standards-compliant way, is imperative. Again, it is important that it be possible to use the same text with a number of different analytical packages and in a number of different environments. In an environment where a well-designed general-purpose package serves a core of needs in a client/server environment, it may be possible that more specialized activities will move to a post-processing phase supported by specialized clients speaking to the central server. By using centralized storage and retrieval from larger bodies of materials, we can make it unnecessary for scholars to circumscribe their perspectives based on textual resources that fit on their desktops.

CAPABILITIES OF PAT: DOES IT MEET THE NEEDS OF TEXTUAL ANALYSIS?

At least one package offers all or most of the capabilities outlined as critical pieces of a generalized textual analysis engine. PAT offers

extraordinary performance in both word and phrase searching over large bodies of material and can take significant advantage of SGML encoding. (Examples that follow are from the still incomplete *English Poetry Database,* currently a corpus of approximately fifty million words. Most of the searches that follow yield results in approximately one second.)

- *Speed:* As mentioned previously, words and phrases, in a body of more than 1,500 volumes, can be retrieved in a second or seconds. Combinations of words or phrases do not take significantly more time (ex.: "lond" yields 1,559 matches; "lond" "near" "home" yields 8 matches).
- *Phrase searching:* PAT employs an unusual indexing scheme (Pat trees) to orient retrieval primarily to phrases or strings rather than words (Gonnet, Baeza-Yates, and Snider 1991). Called "semi-infinite string indexing," the indexing allows the software to retrieve phrases with essentially the same speed as it does words (ex.: "lond of troy" yields 4 matches).
- *Truncation:* Truncation benefits from the same indexing that aids phrase searching. Stems are easily searched; truncation is eliminated by adding a space to the end of the search (ex.: "lond " and "lond" yield 1,559 and 6,738 matches).
- *Stop words:* Although PAT does support the concept of stop words, minimal index overhead (roughly 75 percent of the total size of the text), low costs for disk space, and high retrieval speeds make it possible to index without stop words (ex.: "to be or not to be" yields 15 matches).
- *Structure recognition:* PAT can generate structures based on tags or tag relationships. For example, the structure "stanza" is created by indexing the space between the <stanza> and </stanza> tags. It can also create structures based on composite or abstract features, such as the structure "rhymed," which consists of all poems including the attribute "rhymed=y," or the structure "C16," which consists of the body of all works published in the sixteenth century or whose authors flourished in the sixteenth century. Examples:

  docs poem (i.e., how many poems does the *EPD* include? 64,670)
  docs poem not incl "rhymed=y" (i.e., how many unrhymed poems are there in the *EPD*? 2,346)
  docs poem incl.20 docs stanza (i.e., how many poems in the *EPD* have more than twenty stanzas? 2,812)
  (lond—london) within docs C15 (i.e., excl. London, how many words beginning with "lond" are there in 15c *EPD* poems? 362)

PAT's SGML awareness is flexible. In addition to its ability to index based on tags or tag relationships, PAT is accompanied by a suite of tools extremely valuable in processing texts in SGML. A parsing tool, sgmlregion, can check the SGML validity of a document and can generate

rudimentary display specification rules files to view texts in a variety of ways (Figure 3). A structure indexing tool, multiregion, provides a thorough low-level checking of tags and tagging that discovers errors missed by most true SGML parsers.[11]

PAT's support for CONCUR is probably unintentional and centers around its recognition (not enforcement) of SGML. In the following example, pages and page IDs do not always coincide with poems and stanzas. Note that "page 4" ends in the middle of the second poem.

```
<poem>                                          <page n=4>
<stanza>
<l> From fairest creatures we desire increase,</l>          <l> From fairest creatures we desire increase,</l>
<l> That thereby beauties Rose might neuer die,</l>         <l> That thereby beauties Rose might neuer die,</l>
<l> But as the riper should by time decease,</l>            <l> But as the riper should by time decease,</l>
<l> His tender heire might beare his memory:</l>           <l> His tender heire might beare his memory:</l>
</stanza><stanza>
<l> But thou contracted to thine owne bright eyes,</l>      <l> But thou contracted to thine owne bright eyes,</l>
<l>Feed'st thy lights flame with selfe substantiall fewell,</l>  <l> Feed'st thy lights flame with selfe substantiall fewell,</l>
<l> Making a famine where aboundance lies,</l>              <l> Making a famine where aboundance lies,</l>
<l> Thy selfe thy foe, to thy sweet selfe too cruell:</l>   <l> Thy selfe thy foe, to thy sweet selfe too cruell:</l>
</stanza><stanza>
<l> Thou that art now the worlds fresh ornament,</l>       <l> Thou that art now the worlds fresh ornament,</l>
<l> And only herauld to the gaudy spring,</l>              <l> And only herauld to the gaudy spring,</l>
<l> Within thine owne bud buriest thy content,</l>         <l> Within thine owne bud buriest thy content,</l>
<l> And tender chorle makst wast in niggarding:</l>        <l> And tender chorle makst wast in niggarding:</l>
</stanza><stanza>
<l> Pitty the world, or else this glutton be,</l>          <l> Pitty the world, or else this glutton be,</l>
<l> To eate the worlds due, by the graue and thee.</l>     <l> To eate the worlds due, by the graue and thee.</l>
</stanza></poem>
<poem><stanza>
<l> When fortie Winters shall beseige thy brow,</l>        <l> When fortie Winters shall beseige thy brow,</l>
<l> And digge deep trenches in thy beauties field,</l>     <l> And digge deep trenches in thy beauties field,</l>
<l> Thy youthes proud liuery so gaz'd on now,</l>          <l> Thy youthes proud liuery so gaz'd on now,</l>
<l> Wil be a totter'd weed of smal worth held:</l>         <l> Wil be a totter'd weed of smal worth held:</l>
</stanza><stanza>
<l> Then being askt, where all thy beautie lies,</l>      <l> Then being askt, where all thy beautie lies,</l>
<l> Where all the treasure of thy lusty daies;</l>        <l> Where all the treasure of thy lusty daies;</l>
<l> To say within thine owne deepe sunken eyes,</l>       <l> To say within thine owne deepe sunken eyes,</l>
<l> Were an all-eating shame, and thriftlesse praise.</l>  <l> Were an all-eating shame, and thriftlesse praise.</l>
</stanza><stanza>
<l> How much more praise deseru'd thy beauties vse,</l>   <l> How much more praise deseru'd thy beauties vse,</l>
<l> If thou couldst answere this faire child of mine</l>  <l> If thou couldst answere this faire child of mine</l>
<l> Shall sum my count, and make my old excuse</l>        <l> Shall sum my count, and make my old excuse</l>
<l> Proouing his beautie by succession thine.</l>         <l> Proouing his beautie by succession thine.</l>
</stanza><stanza>                                        </page><page n=5>
<l> This were to be new made when thou art ould,</l>     <l> This were to be new made when thou art ould,</l>
<l> And see thy blood warme when thou feel'st it could.</l>  <l> And see thy blood warme when thou feel'st it could.</l>
</stanza></poem>
<poem><stanza>
<l> Looke in thy glasse and tell the face thou vewest,</l>  <l> Looke in thy glasse and tell the face thou vewest,</l>
<l> Now is the time that face should forme an other,</l>  <l> Now is the time that face should forme an other,</l>
<l> Whose fresh repaire if now thou not renewest,</l>     <l> Whose fresh repaire if now thou not renewest,</l>
```

Figure 2. Flexible views, along with display specification language.

PAT offers a solution to a challenging problem in SGML, a sort of support for CONCUR. Briefly, CONCUR is a feature of SGML designed to support tag relationships that do not nest in a predictable way.[12] For example, pages do not coincide with chapters, always beginning within a chapter, and finishing within a chapter. SGML packages available today rarely support CONCUR. Because PAT and supporting tools can generate structures selectively, rather than necessarily processing all tag pairs in one pass, it is possible to index two conflicting streams of tags in different processes, thereby avoiding the conflict (Figure 3).[13]

PAT's Quiet Mode is a complete language suitable for client-server communication. This sample of communication with the *Oxford English Dictionary* demonstrates a search of a word and then a co-occurrence search, followed by a display of ten sampled hits with 250 characters of context. The dialogue is marked with "Client" communication and "Server" response, and results are numbered, for readability.

```
Client:   "lond "
Server:   <SSize>13210</SSize>
Client:   "lond " near "home"
Server:   <SSize>38</SSize>
Client:   {Quieton raw}; {Printmode 1}; pr.250 sample.10
Server:   <PSet>
1.    <Start>187380797</Start><Raw><Size>250</Size>
      A> <W>Charit. Lond.</W> 31 <T>The Home for Confirmed
      Invalids. </T></Q><Q><D>1863</D> <A>S. Low</A> <W>Charit.
      Lond.</W> Index 312 <T>Home for Aged Annuitants.
      </T></Q><Q><D>1897</D> <W>Whitaker's Alm.</W> 282 <T>Dr.
      Barnardo's Homes for Orphan Waifs
      </Raw>
2.    <Start>26391928</Start><Raw><Size>250</Size>
      D>1723</D> <W>Lond. Gaz.</W> 6127/3 <T>The Mayor..having
      appointed Carew Davis..Pumper of all the Bath-waters.
      </T></Q><Q><D>1836</D> <W>Scenes Commerce</W> 162 <T>The
      Bath water is hot.</T></Q></PQP><PQP><Q><D>1795</D> <A>W.
      Lewin</A> <W>Insects Gt.
      </Raw>
      [six occurrences deleted]
8.    <Start>309187934</Start><Raw><Size>250</Size>
      D>1867</D> <W>Lond. Rev.</W> 22 June 696/1 <T>To restore our
      rivers to their former prolific condition, it is
      indispensable that salmon-passes should be provided.
      </T></Q><Q><D>1899</D> <W>Daily News</W> 4 May 11/2 <T>In
      1863 a salmon pass or ladder
      </Raw>
9.    <Start>48322082</Start><Raw><Size>250</Size>
       <W>Old Home, Lond. Suburb</W> (1879) 244 <T>A calm variety
      of incident.</T></Q></QP></S6></S4><p><S4><#>2</#>
      <S6><DEF><LB>Comb.</LB>, as
      <IL><LF>calm-minded</LF><SF>calm-minded</SF><MF>calm-minded<
      /MF></IL>, <IL><LF>calm-mindedness</LF><SF>-mindedn
      </Raw>
</PSet>
```

Figure 3. Pseudo-CONCUR

PAT also has failings. For example, it does not yet support regular expression searching. That is, single character internal truncation, variable character substitution (e.g., "m[aoe]n" retrieves "man" "mon" and "men") is not yet possible. By indexing for left-hand truncation, one introduces annoying problems in other areas and quadruples index sizes.[14] And the software is supplied without guidance for implementing a service in a campus-wide environment (i.e., the expectation seems to be that every potential user will have an account on the host machine). Still, PAT's design intelligence is one that accommodates many research needs while accommodating needs of long-lived documents, i.e., documents in SGML.

LEVELS OF USE AND WORK PERFORMED: UNIVERSITY OF VIRGINIA AND UNIVERSITY OF MICHIGAN

In 1993, the first full calendar year of use at the University of Virginia, nearly 1,700 University-affiliated persons logged 7,533 total sessions. A session may last only a few minutes or several hours, and may involve one or several databases. Overwhelmingly, sessions were logged by students and faculty from the School of Arts and Sciences. This finding is distinct from that at the University of Michigan, where the second largest user group was from the School of Engineering. The difference between the two universities is explained at least in part by the requirement, at the University of Michigan, for every user to acquire an account on the host machine. At Virginia, the mode of access is much more barrier-free and allows for serendipity. Supporting random, unpredicted use is a critical part of support for a textual analysis system.

The types of uses reported here are anecdotal (conveyed to the author in personal e-mail communication) but are meant to represent the complexity of research supported and the limitations of the current system.

Case 1

A University of Michigan scholar interested in Middle English dialect explored characteristics of the texts through PAT's recognition of structure. "One scribe copied [the *Owl and the Nightingale,* Cambridge], but the language shows that two scribes, with different dialects, copied an antecedent version, and that their work can be identified through his handiwork." E. G. Stanley has defined those sections as lines 1-900 and 961-1183 (being the work of one scribe) and 901-60 and 1184-end (as the work of the second scribe). Using the software and its recognition of structure to define those sections, she was able to save the divisions (under the names C1 and C2) and "contrast the two spelling systems."

Case 2

All of Austen's novels and many of her letters are available through the systems at Michigan and Virginia. A philologist at Michigan leads a discussion group that often uses the collection to look at questions of historical language change. One interesting instance is the passive construction with "being" as in "the house was being built." Late in the nineteenth century, this construction began to displace the earlier preferred construction, "the house was building." Austen's *Sanditon* was completed by another writer, and a comparison of the novel with Austen's other writings finds a clear preference for the passive "[was/were/am/is] being" where it is absent in the works authored solely by Austen.

Case 3

A medievalist at the University of Virginia was asked by a student completing a dissertation on *Piers Plowman* "whether the parallel terms/phrases *Do Wel, Do Bet,* and *Do Best* were ever used as infinitives rather than as simple nominals made from the verb form. The verb form has usually been taken essentially to be imperative when it carries a purely verbal sense. . . ." By searching the A, B, and C texts of *Piers Plowman,* he was able to provide the student with "all the instances of each, each in a ten-line context . . . , and the result was [the student] solving one of the important cruxes of the poem."

Case 4

An Anglo-Saxonist at the University of Virginia was completing an edition of a text and decided he should include those words that occurred only in the text he was editing in a glossary. Using the University's online version of the Old English Corpus, he was able to identify each of these. "[T]hough because grammatical inflection often changes the root form of a word (e.g., man, men) and medieval spelling is variable (e.g., hit, hyt), it was often necessary to do more than one search." This quickly led him to the conclusion that support for regular expressions, e.g., "m[aoe]n*" to yield all words beginning with "man," "mon," or "men," is a critical factor currently lacking in the system. However, he reported that he "got more ambitious than that. The Old English Corpus consists of poetry, prose, and glosses to Latin texts, and these three types of texts employ lexicons that are in some ways specialized. That is, there are words that are used only in poetry, and, somewhat surprisingly, words that are used only in glosses. There are prose words that are not used in poetry, though they are used in glosses. My text was, I was already aware, unusual in that it contained a large number of words that were otherwise attested only in glosses, and not just any glosses, but a particular set of them, glossing Latin texts that my author seemed particularly to like. So I was interested in finding those words that were attested, outside of my text, only in glosses." While the Old English Corpus is marked for three types of text (prose, verse, and glosses), it became clear to the scholar that more in the way of classification will be necessary: "Then some folks would like other groupings. Historians might like to be able to rope off the charters in the same way, for example, or the legal texts; some might like to search just the medical texts."

From the perspective of the institution hoping to provide access to the resources of textual analysis, there is much that is promising. Textual collections are available, and the climate for collaborative development of resources is positive. The standards for defining those

textual resources are mature and well articulated. Software to accommodate many types of analytical work is available. An open systems orientation is possible with some of the incipient clients and the server code written to take advantage of PAT and ARTFL's PhiloLogic. Still, much is lacking.

WHERE FROM HERE?

PAT can serve as a foundation for current efforts and future developments but lacks some important capabilities. It must begin to support regular expression searching. Open Text should supply server code (rather than relying on login-based transactions and personal accounts on the host machine for all users) that is as carefully tested and reliable as PAT itself. It also needs competition from other packages.

A search and retrieval protocol that is aware of document structure is needed. Such a protocol would resemble Z39.50 in facilitating a wide range of standardized operations for heterogeneous clients but would differ from Z39.50 in its ability to exploit complex documents.

A related need is a standard query language. Such a query language would function like Z39.58 (the Common Command Language) and would probably only be used by developers in creating clients. A standards-based query language would have the benefit of supporting the development of clients that would speak in a predictable way with a textual analysis server, making possible queries such as "limit searches to the body of the text that has a header with faulkner in author" or "locate verse groups that contain more than one line with the phrase 'were driven'."

Of course, my conclusions about the requirements for software and my sense of the types of uses are not drawn from a detailed analysis of research habits but instead from work with faculty at Michigan and Virginia. For the most part, this is a necessary compromise in beginning to establish services such as this. We are frequently several steps ahead of our constituencies in understanding the limitations and the possibilities of the technology. We know, for example, that Z39.50 is critical in moving our bibliographic systems forward, but the standard and its client/server operation will mean virtually nothing to those who will benefit most from it. We need to create an environment so that the questions of needs can be asked not in a vacuum but with an understanding of the potential of the technology. A thorough analysis of faculty research needs is still necessary. The environment created at the University of Michigan and the University of Virginia can serve as a foundation for that sort of needs analysis, but I am convinced that the resources offered through current systems in those institutions can also serve as the foundation for effective research today and tomorrow.

APPENDIX

ACCESS THROUGH STRUCTURE

Document Structure

A persistent and fundamental problem of creating and accessing libraries of material on the Internet is the matter of the structure of documents. Documents, whether wholly textual or compound documents, exhibit aspects that we generally call "structure." A monograph is comprised of parts, chapters, or essays. A journal is comprised of volumes, issues, and eventually articles. Articles, essays, and chapters are comprised of sections. Compound documents exhibit similar, though less predictable, features. Structure is widely recognized and is used to frame the meaning of documents. Despite this, paradigms for the transfer of information in a networked environment have either eschewed the notion of structure entirely or have represented rudimentary structures through the filesystem's directory and file paradigm. That is, large, complex documents are passed in their entirety from server to client, or coarse levels of structure in the document are modeled by fragmenting the document into directories and files. Both models of document transfer are inadequate for a variety of reasons. Current network capacity often fails to provide sufficient bandwidth to support fluid transfer of large documents; current workstation capacity is frequently overwhelmed by documents of even moderate size. And while both network capacity will increase and workstation capabilities will develop, the user of large documents is not able to navigate easily in these large bodies of text and will frequently want small, well-defined portions. The other alternative currently being used by Gopher and HTML/HTTP, fragmentation of documents into directories and files, is equally untenable, as we build large collections of documents and face the need to transfer even more precisely defined subsections of documents (e.g., entries in a large glossary). An effective means of recognizing and transferring structural features of documents is perhaps the most significant impediment to making large bodies of material available in a networked environment.

Most other problems of access to digital libraries depend on the more fundamental problem of developing a model of access to structured information.[15] The resolution of issues such as copyright, accounting, authentication, and data redundancy will continue to leave unresolved this problem of effective access to documents. Several solutions to these other problems have been proposed, tested, and some are in wide use. Billing servers and model copyright and use agreements such as the Coalition for Networked Information (CNI) Project READI have been offered as possible solutions to these problems. Kerberos and the corresponding Distributed Computing Environment (DCE) implementation of this authentication scheme may be a wholly adequate means of assessing the identity and rights of a user querying a large document server. However, without a resolution of issue-structured access, none of these proposed solutions can be widely used or tested.

Related Standards

The NISO standard, Z39.50, promises to revolutionize access to bibliographic databases. Through the standardization of the organization of bibliographic information and the protocols through which client and server

communicate, we have begun to see the disappearance of traditional models of host-based access to bibliographic information. Those traditional models made inevitable the loss of the amenities of the local computing environment, amenities such as a sophisticated user interface and reasonably easy capture of information. Systems for retrieval have proliferated, despite the promulgation of a Common Command Language (Z39.58). In a very short time, graphical user interfaces for database navigation, searching, and information retrieval have been developed. Similarly, intercommunication between dissimilar systems has begun to take place, making differences between systems less significant and the choice of a user interface the governing factor.

A relatively simple extension of Z39.50 from a bibliographic environment to document retrieval is unlikely to take place. We will not be able to force all documents into a similarly well-defined structural representation: unlike a document, the bibliographic record is one-dimensional and has a predictable and relatively small set of possible fields. Similarly, the dimensions of the bibliographic record do not pose the challenges that are posed by a document. Nesting of elements (e.g., the third section of the first article within the second issue of the fifth volume) is a defining feature of the document. Because such an extension of Z39.50 to the document is unlikely, a similar protocol designed around the needs of the document is necessary. At the same time, we can see in recent developments around Z39.50 some of the great promise held for the establishment of such a protocol.

A Model of Structured Access

A model of access to and transfer of structured documents must be developed. The model should utilize information about the structure carried explicitly in a statement of the document's grammar. Each document might declare conformance to a specific Document Type Definition (DTD) in the interaction between client and server. The statement of conformance would be passed from client to server upon initial request for a document by the client. DTDs minimally convey information about the constituent parts of a document, and the relationship between those parts within the document. In the first interaction between client and document, the client browser will offer a virtual table of contents based on the highest level(s) of structure reflected in the DTD. A book DTD might offer the choices of parts, chapters, and sections, arrayed hierarchically in the browser. A more generic DTD might express those options as subdivisions of the body of the text where level one divisions (e.g., "div1") would be subdivided by level two divisions (e.g., "div2").

DTDs can be registered with the ISO, expediting this process. Built into the client would be the basic structural components of most widely used DTDs, including those from the American Association of Publishers (AAP), the American Mathematical Society (AMS), and the TEI.[16] In most of these cases, structure names will be resolved in the browser as natural-language names (e.g., "Chapter" rather than "chap"). In other cases, such as the more generic situations that prevail in TEI DTDs, generic structural divisions will be expressed by their tag names with available attribute values and will be arrayed in a similar hierarchical subdivision to facilitate browsing.

User requests would be mediated by the client in order to retrieve the relevant structure's contents. Upon being presented with a structure map or virtual table of contents, two operations would be immediately possible. By selecting one portion of the browser (e.g., a bullet opposite an entry), the server would

provide the client with a list of substructures of the selected structure as found in the document. By selecting the name of the structure (e.g., "Chapter 1"), the contents of the structure will be passed to the user's client. Should the individual making the request desire to receive the contents of the entire document, this will also be possible. In this way, efficiencies of network and workstation capacity and reader ability will be maintained.

A Structure-Aware Query Language

Just as Z39.50 benefited from the Common Command Language (Z39.58), the proposed representation of structure can benefit from the articulation of a query language that is structure aware. Such a language must be capable of expressing nested relationships and must be capable of distinguishing between structures and text strings. So, for example, the language must be able to respond both to "give me the bibliography of the third chapter in part two" and "show me articles in the Journal of X where the author uses the phrase 'the indeterminacy of language'." A standards-based query language would make it possible for a developer to create a client that could speak in a predictable way with a textual analysis server, thereby effectively encouraging the development of many different types of clients supporting many different functions. These might range from simple browsers of electronic text to specialized post-processing clients where lemmatization or statistical analysis might take place based on locally defined rules of no consequence to the server or its search engine.

NOTES

[1] PAT is available from Open Text Corporation, 180 King Street South, Suite 550, Waterloo, Ontario N2J 1P8, Canada. Tel.: 519-571-7111; Fax: 519-571-9092.

[2] It is not my intention to discuss the hardware needed to make such a service available, as the options available have changed so substantially in such a short period of time. The RS/6000 Model 320 Michigan first used to make its service available cost approximately $15,000, while disk drives cost nearly $3,000 per gigabyte. Currently, a similar (and still appropriate) server costs less than half that of the 1989 Model 320, and disk drives cost approximately $750 per gigabyte. RAM continues to be a disproportionately large part of the cost: while 32Mb is satisfactory, in order to support large indexing and many simultaneous users, 64Mb is more appropriate. Any of a number of workstation class UNIX computers is well suited to this task.

[3] Project Mercury is described in Mark Kibbey and Nancy H. Evans. 1989. The Network Is the Library. *EDUCOM Review* 24(3): 15-20. The TULIP project has been discussed in Karen Hunter and Jaco Zijlstra. 1994. TULIP the University Licensing Project (for delivery and use of journals). *Journal of Interlibrary Loan, Document Delivery & Information Supply* 4(3-4): 19-22.

[4] The vt100 client was developed by Yuzhen Ge and John Price-Wilkin, working from a model of menuing in other Open Text applications and elaborating code written by Paul Pomes at the University of Illinois at Urbana-Champaign. The software is provided freely and without support. For more information about retrieving the software, please see the file "announce," available via anonymous ftp in the "pub" directory at etext.virginia.edu.

[5] A careful construction of this search would eliminate false drops. In this example, only *London* is removed from the "lond" set. The searches used are:

```
docs body incl (lond—london)
docs body incl land
1 ∧ 2
(lond—london) within 3
```

[6] The first preliminary draft of the guidelines was published in 1990. The second draft, known as P2, has been published subsequently in fascicles and is available via anonymous FTP from file://sgml1.ex.ac.uk/tei/p2. A complete, revised edition (P3) was published in 1994 (Association for Computers and the Humanities et al. 1994).

[7] Triggs is the Director of the Oxford English Dictionary's North American Reading Program and can be reached at triggs@bellcore.com.

[8] While the Old English Corpus is not available via anonymous FTP, most of the Oxford Text Archive's SGML-conformant materials are from black.ox.ac.uk, in the ota directory.

[9] Of course, many works of English verse will not be included in the *English Poetry Database*. Chadwyck-Healey's source, the *New Cambridge Bibliography of English Literature* is not exhaustive for the period covered, works published only in periodicals will probably not be included, and works published in the twentieth century are omitted entirely.

[10] It is also the case that scholars have found these resources important in their work. I believe it is because there is such intrinsic quality in many of the resources (from both good and bad editions) and because they are educated users of the resources, using caution where caution is appropriate.

[11] Multiregion reports occurrences of "mangled tags" such as the double "<<" at the end of the title: "<header><fileDesc><title>The Feasibility of . . . <</title> ." The double "<<" may be valid content, according to the Document Type Definition (DTD), but is clearly a mistake.

[12] This is an admittedly superficial description of CONCUR. At one point, Goldfarb (1990, 177) describes CONCUR as a "feature of SGML, which allows instances of multiple document types to exist concurrently in the same document."

[13] PAT also provides a sort of SGML in its communication with other programs. PAT communicates (for example, through client/server relationships) using what it calls

"Quiet Mode." PAT's Quiet Mode is a structure-aware command syntax to report all results in paired tag sets, making possible a more reliable communication between interface and index (Figure 3).

[14] Indexing for left-hand truncation with PAT is problematic. While with texts indexed for right-hand truncation, the user can "turn off" truncation by including a space, as in "lond " it is not possible to search for " lond " when texts are indexed for left-hand truncation. All initial spaces are removed, so that both " lond" and "lond" yield the same results.

[15] The structural representation of documents, both compound and flat, has largely been achieved by recent standards developments (ISO 8859, ISO 10744, and ISO 10646). ISO 8859, or SGML, has remained virtually unchanged since its passage in 1986 and continues to be a valuable resource in this area. In 1992, the ISO passed both HyTime, a standard for time-based compound or multimedia documents, and ISO 10646, or UCS, a 16-bit character encoding scheme for the world's alphabets. Specific implementations of these standards—i.e., DTDs, in most cases—are still needed in some areas, but efforts by associations and publishers have largely solved this problem, leaving only the need to elaborate specialized, project-specific DTDs.

[16] The AAP DTDs are widely used in publishing and offer markup guidelines for books, journal articles, tables, and formulas. A variant of the AAP DTDs has recently been approved by the ISO as a more general set of publishing DTDs. The AMS DTDs are probably the most widely used scientific DTDs. The TEI DTDs are being elaborated now but consist of a wide array of tag sets for a variety of applications. They are unquestionably the broadest and most versatile DTDs available for electronic publishing.

REFERENCES

Association for Computers and the Humanities (ACH), Association for Computational Linguistics (ACL), and Association for Literary and Linguistic Computing (ALLC). 1994. *Guidelines for Electronic Text Encoding and Interchange,* TEI P3. Ed. C. M. Sperberg-McQueen and Lou Burnard. Chicago; Oxford: Text Encoding Initiative.

Brentrup, Robert J. 1993. Building a Campus Information Culture. *Cause/Effect* 16(4): 8-14.

Coombs, James H., Allen H. Renear, and Stephen J. DeRose. 1993. Markup Systems and the Future of Scholarly Text Processing. In *The Digital Word: Text-Based Computing in the Humanities,* ed. George P. Landow and Paul Delany, 85-118. Cambridge, Mass.: MIT Press.

Goldfarb, Charles F. 1990. *The SGML Handbook.* Oxford, Eng.: Clarendon Press.

Gonnet, Gaston H., Ricardo A. Baeza-Yates, and Tim Snider. 1991. *Lexicographical Indices for Text: Inverted Files vs. Pat Trees.* Waterloo, Ont.: UW Centre for the New Oxford English Dictionary.

Price-Wilkin, John. 1991. Text Files in Libraries: Present Foundations and Future Directions. *Library Hi Tech* 9(3): 7-44.

Price-Wilkin, John. 1992. A Campus-wide Textual Analysis Server: Projects, Prospects, and Problems. In *Proceedings of the Eighth Annual Conference of the UW Centre for the New OED and Text Research.* Waterloo, Canada: UW Centre for the New OED and Text Research.